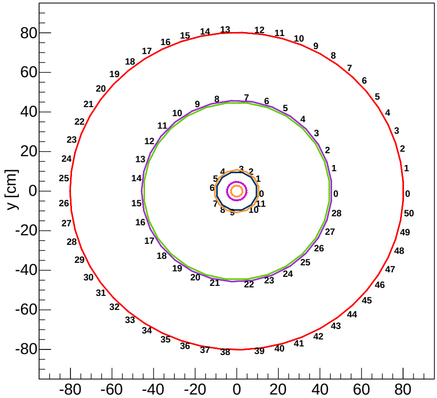
Tracking Requirements

- Hard requirements
  - 100 MeV/c^2 Υ mass resolution
  - DCA resolution sufficient for b-tagging
- Softer requirement
  - high p_T resolution for particle flow



Layer	radius (cm)	sensor pitch (μm)	sensor length (mm)	sensor depth (μm)	total thickness % X_0	area m^2
1	2.7	50	0.425	200	1.3	0.034
2	4.6	50	0.425	200	1.3	0.059
3	9.5	60	8	320	1.35	0.152
4	10.5	240	2	320	1.35	0.185
5	44.5	60	8	320	1	3.3
6	45.5	240	2	320	1	3.5
7	80.0	60	8	320	2	10.8

7 layers of silicon, really 5 layers with 2 of them being double-sided
These are implemented as cylinders by default, though a "ladder" implementation also exists

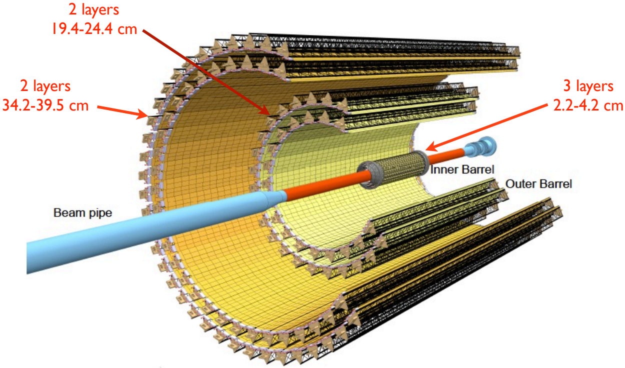
ITS Tracker

All layers composed of pixels.

- Inner three layers: 0.3% / layer
- Outer four layers: 0.8% / layer

Total thickness $X/X_0 = 4.1\%$

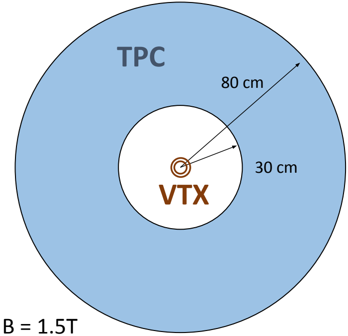
Pixel sizes:
inner barrel 20-30 x 20-30 μm
outer barrel 20-50 x 20-50 μm



replace the pixels with 3 layers of ALICE MAPS technology

A sPHENIX Tracking Solution: TPC & 2 Pixel Layers

Design parameters and performance consistent with ILC TPC prototypes



TPC

- T2K gas volume from 30cm to 80cm
- $|y|< 1 \rightarrow z_{max} \sim \pm$ 80cm
- 60 radial readout layer with $\Delta r \sim$ 8mm
- Readout plane with 1.2mm pads in $r\varphi$
  - approximately 350,000 readout channel
- Assume 40 MHz FADC $\rightarrow$ $\Delta z \sim$ 2mm
  - approximately 400 samples per readout channel

VTX layers 1 & 2

- Silicon tracker design

Motivation : better Υ resolution, low momentum PID, could be day 1 EIC tracker?

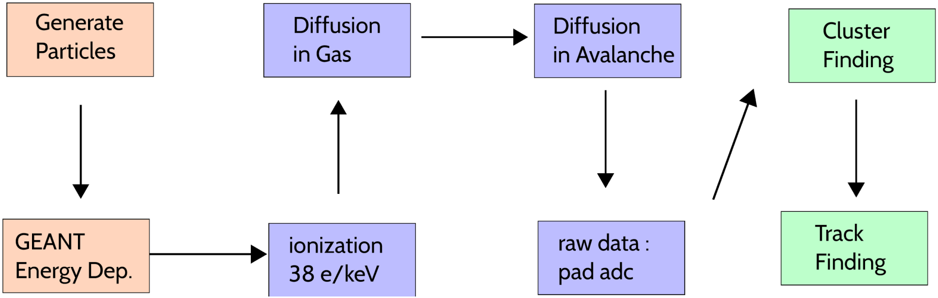
Lots of work to be done still : space charge effects for example have not been implemented at all.

High level view of tracking framework

- GEANT just gives us energy deposits in material. The silicon trackers are implemented as active silicon and inactive copper behind it
- For the silicon, the layers are logically divided into pixels. Charge from a track is distributed into pixels based on the entry and exit point of the track in the active material.
- The hit pixels are then clustered together (using ADC information in the strips).
- Pattern recognition is done using a hough tranform, and track fitting is done with a Kalman filter
- The pattern recognition currently only works in a homogeneous field. More than about 10% distortions from uniform causes track-finding to break down. So this can't be used for forward tracking. Do we want a situation like in current PHENIX where the central arm and muon code are mostly separated?

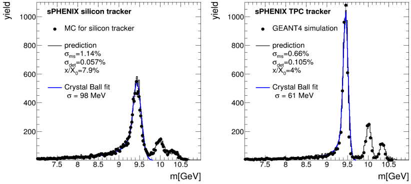
TPC flowchart

Flow Chart of GEANT Simulation & Analysis

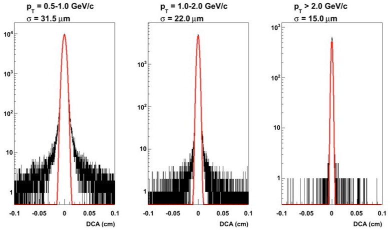


Need to stretch radius to meet our momentum resolution spec
Consider 20x20 and 20x20 μm pixel size
Total thickness = 4.1%

GEANT4 Simulation of Upsilon Measurement



DCA resolution 60 cm outer layer



The track kinematics/trajectory are stored in the output file. Currently by default the trajectory is stored near the vertex. The outward trajectory should be stored as well. The covariance matrix of all track parameters is available. See the file

`https://github.com/sPHENIX-Collaboration/coresoftware/blob/master/simulation/g4simulation/g4hough/SvtxTrack.h`

Output

Everything works nicely for making plots for the proposal with little manpower
However, now that we have a collaboration forming,

we should define a stable analysis interface

Reconstruction can change, but if we will have many people really using the software, the analysis API needs to be stable. Part of the discussion should be defining a clean, precise interface that satisfies the needs of all potential analysis at least up to the conceptual design.

Side note : we should also define what parts of the g4"main" are considered stable (and binary compatible).

Side side note : we should stop assuming access to RCF

Code organization

All the code related to detector "hit" information is in the `g4detectors` directory. Clustering and track finding/fitting are in `g4hough`.

Note : currently all code for the TPC is in the "TPC" branch on github.

Macros are at
`https://github.com/sPHENIX-Collaboration/macros/tree/master/macros/g4simulations`

`Fun4All_G4_sPHENIX.C` is a macro to steer the analysis. It is pretty self explanatory. One has to go in and modify some boolean flags to change the input file format and what detector are simulated

`G4Setup_sPHENIX.C` defines the magnetic field map and which detector specific macros to load. For example, modify
```
gROOT->LoadMacro("G4_Svtx.C");
```
to
```
gROOT->LoadMacro("G4_Svtx_ITS.C");
```
to use the ALICE ITS pixels

Workfest Topics

That's for all of us to decide, but here are some ideas

- Silicon tracker nitty-gritty
  - Kalman filter improvements for tilted, double-layered strips
- TPC tracker nitty-gritty
  - Space charge distortions
  - dE/dx PID study
- Physics analysis!
  - D meson reconstruction in HIJING
  - does particle flow jet-finding help in HIJING?
  - Upsilon reconstruction...wait for it...in HIJING!

We can design the detector for kinematic resolution on a piece of paper, but complicated full analyses can find potential weak spots